50 Medium–Hard MCQs on Software Testing (Categorized)

Each question includes four options (a–d), the correct answer, and a brief explanation.

Fundamentals & SDLC

Q1. Which statement about verification vs. validation is TRUE?

a) Verification ensures the product meets user needs; validation checks specs.

b) Verification is dynamic; validation is static.

c) Verification asks 'Are we building the product right?'; validation asks 'Are we building the right product?'

d) Both require execution of the software.

Answer: c

Explanation: Verification is about conformance to specifications (build it right); validation is about meeting user needs (build the right product).

Q2. In the V-Model, component testing aligns most closely with which development activity?

a) System design

b) Component design

c) Requirements analysis

d) Acceptance design

Answer: b

Explanation: Component/unit testing verifies individual components against their detailed/component design.

Q3. In Agile teams, which practice best reduces defect leakage across sprints?

a) Big-bang testing

b) Sprint hardening only

c) Definition of Done with unit tests and clear acceptance criteria

d) Post-release patching

Answer: c

Explanation: A robust DoD (tests, code review, acceptance criteria) embeds quality in each sprint.

Q4. Exit criteria for a test phase typically include all EXCEPT:

a) Target coverage achieved

b) No open high-severity defects

c) All planned tests executed

d) Budget completely consumed

Answer: d

Explanation: Budget consumption is not a quality-based exit criterion.

Q5. Which is the MOST suitable basis for risk-based test prioritization?

a) Newest features first

b) Likelihood × impact of failure

c) Developer preference

d) Alphabetical order of modules

Answer: b

Explanation: Risk is commonly assessed as likelihood times impact; higher risk → higher priority.

Static Testing (Reviews, Inspections, Static Analysis)

Q6. During a formal inspection, who is primarily responsible for logging defects?

a) Moderator

b) Scribe

c) Author

d) Reviewer

Answer: b

Explanation: The scribe records defects and action items; the moderator facilitates.

Q7. Which artifact is LEAST suited for automated static analysis tools?

a) Source code

b) Requirements specification document

c) Test scripts

d) Configuration files

Answer: b

Explanation: Automated static analysis targets code-like artifacts; natural-language requirements need reviews rather than static code analysis.

Q8. A key benefit of static testing is:

a) Measuring response time under load

b) Early defect detection and reduced rework

c) Validating deployment pipelines

d) Generating throughput graphs

Answer: b

Explanation: Static testing finds defects early without execution, cutting rework cost.

Q9. A common quantitative output of review activities is:

a) Executable test results

b) Defect density metrics and rework items

c) Memory leak reports

d) SLA conformance charts

Answer: b

Explanation: Reviews can produce metrics like defects per page/function and rework actions.

Test Design Techniques — Black-Box

Q10. For input range 1–100 inclusive, which is an INVALID equivalence class?

a) 1–100

b) ≤ 0

c) 50–75

d) 1–50

Answer: b

Explanation: Values ≤ 0 fall outside the valid range.

Q11. For a field that accepts 0–255 inclusive, which BVA set is appropriate?

a) −1, 0, 255, 256

b) 0, 1, 254, 255

c) 0, 255

d) 1, 254

Answer: a

Explanation: Classic BVA uses values just below/at/just above each boundary.

Q12. Decision tables are MOST suitable when:

a) Sequential states drive behavior

b) Complex combinations of business rules exist

c) Code structure is known

d) UI aesthetics are the focus

Answer: b

Explanation: Decision tables excel at covering rule combinations and their outcomes.

Q13. State transition testing is primarily used to test:

a) CPU utilization

b) Memory leaks

c) Behavior depending on event history

d) Sorting correctness

Answer: c

Explanation: State behavior depends on prior events/transitions, not just current inputs.

Q14. Pairwise (2-way) testing is based on the assumption that:

a) All N-way interactions are equally likely

b) Most defects are triggered by interactions of two factors

c) It replaces boundary testing

d) It requires code access

Answer: b

Explanation: Empirical evidence shows many defects arise from 2-way interactions.

Q15. You must test operators {+, −, ×, ÷} with positive/negative operands while minimizing cases. Which technique is BEST?

a) Full decision table of all operand ranges

b) Pairwise across operator and operand signs

c) State transition testing

d) Path coverage

Answer: b

Explanation: Pairwise reduces combinations yet covers each pair of factor values.

Q16. For validating Bangladeshi NID input (10 or 13 digits with checksum), which mix is MOST effective?

a) Equivalence classes + BVA + error guessing

b) Statement coverage

c) Spike testing

d) Fault injection

Answer: a

Explanation: Combine partitioning, boundaries, and heuristic invalids to hit validation logic.

Q17. For file uploads (≤10 MB, types PDF/JPG), which design ensures each constraint is tested at least once?

a) Pairwise between file size and type

b) State-based testing

c) Mutation testing

d) Fuzz testing only

Answer: a

Explanation: Pairwise guarantees coverage of size/type combinations efficiently.

White-Box Techniques & Coverage

Q18. Which coverage metric ensures both outcomes (True/False) of each decision execute at least once?

a) Statement coverage

b) Decision/branch coverage

c) Condition coverage

d) Path coverage

Answer: b

Explanation: Decision (branch) coverage requires each decision to take both outcomes.

Q19. Which coverage generally subsumes statement coverage in structured code?

a) Decision/branch coverage

b) Condition coverage

c) Path coverage

d) Function coverage

Answer: a

Explanation: Exercising branches will typically execute all statements too.

Q20. MC/DC coverage requires that:

a) All paths execute

b) Each condition independently affects the decision outcome

c) Each statement runs at least once

d) Each loop iterates to its max

Answer: b

Explanation: Modified Condition/Decision Coverage isolates each condition’s effect.

Q21. Typical loop testing includes which set of iterations?

a) 0, 1, 2, N−1, N, N+1

b) 1, N

c) 0, N

d) N only

Answer: a

Explanation: Test zero, one, two, typical, max, and just-beyond limits.

Q22. In unit testing, how is dependency isolation BEST achieved?

a) Service virtualization / mocks/stubs

b) Equivalence partitioning

c) Statement coverage

d) Fuzzing

Answer: a

Explanation: Mocks/stubs (or service virtualization) isolate the unit from external dependencies.

Q23. Mutation testing evaluates:

a) System performance under spikes

b) The fault-detection strength of a test suite by seeding faults

c) User acceptance criteria

d) Security posture against OWASP Top 10

Answer: b

Explanation: Mutants simulate typical coding faults; surviving mutants indicate weak tests.

Levels & Types of Testing

Q24. A key drawback of big-bang integration testing is:

a) Simplicity

b) Early fault isolation

c) Difficult fault localization

d) Need for stubs/drivers

Answer: c

Explanation: When everything is integrated at once, isolating failures is difficult.

Q25. System testing primarily focuses on:

a) Individual units

b) Interfaces among a small set of modules

c) Entire integrated system vs. requirements

d) Customer’s live environment

Answer: c

Explanation: System testing validates the complete integrated product against requirements.

Q26. The PRIMARY goal of acceptance testing is to:

a) Verify code coverage

b) Validate business needs and readiness for release

c) Find memory leaks

d) Optimize database queries

Answer: b

Explanation: Acceptance testing assures the product meets business needs and is releasable.

Q27. For microservice architectures, which approach is MOST appropriate?

a) Top-down integration only

b) Contract testing

c) Big-bang integration

d) Unit testing only

Answer: b

Explanation: Contract tests verify service interface expectations independently of full integration.

Q28. Which testing ensures environment/configuration properties are correct after deployment?

a) Sanity testing

b) Smoke testing

c) Configuration testing

d) Regression testing

Answer: c

Explanation: Configuration testing validates environment variables, properties, and dependencies.

Test Management, Defects & Metrics

Q29. Which belongs more to a TEST STRATEGY than a test plan?

a) Specific schedules

b) Staffing for this release

c) Organization-level approaches and standards

d) Build numbers

Answer: c

Explanation: Strategy is higher-level, reusable across projects; plans are project-specific.

Q30. A requirements traceability matrix (RTM) links:

a) Modules to classes

b) Requirements to test cases and defects

c) Developers to tasks

d) Environments to builds

Answer: b

Explanation: RTM ensures coverage from requirements through tests and recorded defects.

Q31. Which statement about severity vs. priority is TRUE?

a) Severity is business urgency; priority is technical impact

b) Severity is technical impact; priority is business urgency

c) They are identical

d) Only testers set both

Answer: b

Explanation: Severity reflects technical impact; priority reflects how soon to fix (business need).

Q32. A defect is set to 'Deferred' when:

a) It cannot be reproduced

b) It is fixed and verified

c) It is valid but scheduled for a later release

d) It is rejected as 'not a bug'

Answer: c

Explanation: Deferred = acknowledged but postponed.

Q33. Which metric is MOST indicative of test effectiveness?

a) Lines of code tested per hour

b) Defect detection percentage (DDP)

c) Number of testers

d) Code churn

Answer: b

Explanation: DDP = defects found during testing ÷ total defects (testing + post-release).

Q34. The MOST appropriate basis to stop testing is:

a) All tests pass

b) Exit criteria met and residual risk acceptable

c) No open defects at all

d) Budget is exhausted

Answer: b

Explanation: Stopping is based on agreed exit criteria and acceptable risk, not absolute 'no defects'.

Q35. In risk assessment, risk is commonly computed as:

a) Probability + complexity

b) Likelihood × impact

c) Impact − mitigation

d) Likelihood ÷ exposure

Answer: b

Explanation: Risk = likelihood of failure × impact if it occurs.

Automation & CI/CD

Q36. Which tests are the POOREST candidates for automation?

a) Stable, repeatable regression tests

b) Highly UI-volatile exploratory tests

c) Data-driven API tests

d) Smoke tests

Answer: b

Explanation: Exploratory and rapidly changing UI flows are fragile and costly to automate.

Q37. A key benefit of BDD (e.g., Gherkin) is:

a) Faster compilation

b) Shared understanding via executable specifications

c) Fewer unit tests needed

d) Eliminates testers

Answer: b

Explanation: BDD scenarios bridge business and tech by making behavior explicit and testable.

Q38. Flaky tests in CI pipelines are BEST addressed by:

a) Rerunning until they pass

b) Disabling them

c) Root-causing nondeterminism and stabilizing waits/mocks

d) Ignoring them

Answer: c

Explanation: Fixing timing, data, and dependency issues removes flakiness sustainably.

Q39. The Page Object pattern primarily helps by:

a) Maximizing code coverage

b) Decoupling test logic from UI locators

c) Increasing server throughput

d) Encrypting test data

Answer: b

Explanation: It centralizes element locators and actions, reducing maintenance.

Q40. In API testing, what should be asserted FIRST before body validations?

a) Status code and Content-Type

b) Response body length

c) Presence of Server header

d) Number of cookies

Answer: a

Explanation: Correct status and MIME type are prerequisites to meaningful body assertions.

Performance & Reliability

Q41. Soak (endurance) testing differs from stress testing because it:

a) Pushes the system beyond capacity

b) Runs at normal/peak load for long duration to find leaks

c) Focuses on single-user latency

d) Validates only peak concurrency

Answer: b

Explanation: Soak looks for issues like leaks/slow degradation over time.

Q42. Throughput is PRIMARILY measured as:

a) Requests per second

b) 95th percentile response time

c) CPU utilization

d) Memory footprint

Answer: a

Explanation: Throughput is a volume/time metric; latency percentiles complement it.

Q43. A 95th percentile response time of 800 ms means:

a) Average response time is 800 ms

b) 95% of requests are 800 ms or faster

c) 5% of requests are 800 ms or faster

d) 95 requests took exactly 800 ms

Answer: b

Explanation: Percentile means 95% of samples fall at or below 800 ms.

Q44. Bottleneck analysis aims to identify:

a) Cosmetic UI defects

b) The slowest resource limiting throughput

c) Number of test cases

d) Security flaws

Answer: b

Explanation: The limiting resource (CPU, DB, disk, network) caps throughput/latency.

Q45. BEST practice before running load tests in a shared environment is to:

a) Disable monitoring

b) Ensure test data reset and isolation from production traffic

c) Run only during business hours

d) Ignore SLAs during tests

Answer: b

Explanation: Prevent collateral impact and ensure clean, reproducible conditions.

Security & Privacy

Q46. Testing for SQL injection primarily involves:

a) Static testing

b) Dynamic security testing with malicious inputs

c) Performance testing

d) Usability testing

Answer: b

Explanation: Inject crafted payloads at runtime to observe improper query handling.

Q47. Which vulnerability is MOST directly assessed by session management tests?

a) CSRF and session fixation

b) Buffer overflow

c) Command injection

d) Broken file permissions

Answer: a

Explanation: Session tests check token binding, rotation, CSRF defenses, fixation prevention.

Q48. Regarding PII in application logs, testers should verify that:

a) PII is echoed for audits

b) PII is masked or not persisted unless required

c) PII is compressed for storage

d) Logs are always CSV

Answer: b

Explanation: Privacy by design: avoid logging PII or mask/redact it per policy/regulation.

Q49. Testing aligned with OWASP Top 10 'Broken Access Control' focuses on:

a) Rate limiting only

b) Authorization checks across roles/resources

c) TLS version negotiation

d) Cookie SameSite flags

Answer: b

Explanation: Verify users cannot access actions/resources beyond their privileges.

Q50. The PRIMARY value of a threat model to a tester is to:

a) Replace test cases

b) Identify high-risk attack surfaces and prioritize tests

c) Reduce overall test coverage

d) Decide log formats

Answer: b

Explanation: Threat modeling guides risk-focused test design and prioritization.